A Generative Constraint Model for Optimizing Software Deployment

Mihai Nica
Bernhard Peischl
Franz Wotawa

Institute for Software Technology
Graz University of Technology

July 21, 2008

Index

Motivation

- More and more functions in today's cars involve electronics and software
- 80-90 percent of the new innovative features are realized by distributed embedded systems
- Today's upper class cars contain up to 80 ECUs (Electronic Control Units)
- Even highly safety critical mechanical and hydraulic control systems will be replaced by electronic components

Software Deployment

- Today's embedded automotive software is highly distributed
- The automotive industry devotes increasing efforts to develop tools for automated software deployment.
- The underlying foundations comprise techniques like genetic algorithms and clustering techniques.
- Measuring the quality of the deployment process: Metrics

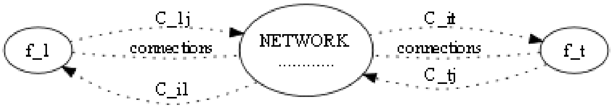
Metrics

Measure the quality of the deployment process

- Bus-load: The system's ECUs are connected by means of data-pipes or buses.
  - It is a critical metric of the system. It's value should be as low as possible. Otherwise: bottlenecks.
- Resources: The number and type of ECUs required to implement the functionalities.
- Costs: The required financial effort for a configuration.

Function Blocks

- Function block: a specific task from a software functionality of the system.
- Example: The Signaling Function
  1. Hit the commutator (Change State)
  2. Select the targeted component (Observer)
  3. Start blinking
- The function blocks have to exchange data ⇒ The network of function blocks.
- Each branch from the network has a weight,the communication frequency *(CF)*.



Clusters

- Cluster $=$ a set of function blocks grouped together.
- To each cluster corresponds an ECU for execution.
- Clusters should contain optimal function distribution.
- Apply quality criteria in order to create the best clustering schema.

Constraint Satisfaction Problem

- A *Constraint Satisfaction Problem* (CSP) is a tuple (V, D, CO).

1. `{ var_1 = ( x_0 < y_0 );`
2. `min_1 = x_0;`
3. `min_2 = y_0; }`

Variables: $V = \{var\_1, x\_0, y\_0, min\_1, min\_2\}$

Domains: $D = \{D(x) = N | x \in V\}$

Constraints:

$$CO = \left\{ \begin{array}{c} var\_1 = (x\_0 < y\_0), \\ min\_1 = x\_0, min\_2 = y\_0 \end{array} \right\}$$

- State of the art constraint solvers are available for solving CSPs.

CSP Partitioning

- The partitioning problem: How to group the function blocks into clusters?
- Transform the problem of cluster partitioning into a CSP:
  - Use the available resources
  - Use Quality Functions
  - Use the cost limitations

Resource Parametrization

- Conversion to CSP $\Rightarrow$ defining the tuple (V, D, CO)
  - Function blocks, $F = \{f_1 ... f_t\}$.
  - The communication frequency sets $CF = \{CF_1 ... CF_t\}$, $CF_i = \{cf_{i1}, ... cf_{it}\}$.
  - Available set of ECUs $ECU = \{ECU_1 ... ECU_k\}$ characterized by:
    - Memory: $MEM = \{mem_1 ... mem_k\}$
    - Processing power: $PROC = \{proc_1 ... proc_k\}$

Resource Constraints

1. The overall memory consumption of the function blocks is smaller or equal to the available memory:
2. Maximal function block memory constraint
3. Each ECU must have enough resources to accommodate the assigned cluster
4. A function block is deployed on a single ECU only
5. Any function *deploy* that distributes all functional blocks f_i on *max* ECUs is a solution.

Timing Constraints

- We assume that:
  - There is only a single path from the source x to the sink y
  - That there are no loops in the network.
- Then we must have:
  1. $t_{xy} \leq \sum_i t_{wc}(f_i) + t_{wc}(B_i) + t_{wc}(G_i)$.
  2. $SF\colon TSK = \{t_1, ..., t_k\} \iff \sum_i time(t_i) \leq T_{SF}$
  3. t_{ECUi}: the time that a ECU needs to execute a task t_i then

$$t_{ECUi} \leq time(t_i).$$

- For every timing requirement $Req_{(a,b)}$ we instantiate these constraints.

Quality Constraints

- The most important factor when we partition the function blocks into clusters.
- We use predefined quality functions.
- Constraint System building: output values that no cluster is allowed to exceed.
- Each quality function receives as input parameter the *CF* set.

Quality Functions

- The External-Internal Ratio: ratio between the external and the internal costs must be as low as possible.

$$\forall C_i, i \in [1, c] \; \frac{E_i}{I_i}$$

- The Davies Bouldin Criteria: shows a good partitioning when the factor is as low as possible.

$$DB = \frac{1}{c}\sum_{i=1}^{c} max_{j \neq i} \left[\frac{diam(C_i) + diam(C_j)}{d(C_i, C_j)} \right]$$

- The Modularization Factor: indicator of a compact clustering. Should be as high as possible.

$$MF = \frac{\sum_i I_i}{\sum_i \frac{N_i(N_i - 1)}{2}} - \frac{\sum_{i<j} E(C_i, C_j)}{\sum_{i<j} N_i N_j}$$

- The SILHOUETTE factor: correctness of the distribution of a function f_i within a cluster C_i with respect to a neighbor node C_j. A good factor is close to 1 :

$$Sh(f_i) = \frac{d(f_i,C_j)-d(f_i,C_i)}{max(d(f_i,C_j),d(f_i,C_i))}$$

- The Cluster Load Deviation: as low as possible. In a good partitioning similar number of function blocks in all clusters:

$$CLD = \sqrt{\frac{1}{c-1}\sum_{i=1}^{c}(N_i - \bar{N})^2}$$

Cost Constraints

- Built based on the system's cost criteria.
- Each ECU has a price and a performance description associated to it.
- The Price Constrain: Distribute the function block set such that the total cost of the ECUs is smaller than a given price.
- The Bus Load Constraint: At any given moment the bus load should not exceed a critical imposed value.

$$CSP = RCS \cup QCS \cup CCS \cup TCS.$$

Generative Deployment of Clusters

- Different types of ECUs on which we must assign the clusters.
- The requirements of this cluster are generated by means of the generative constraints.
- Choose from the set of ECUs the cheapest ECU and try to deploy the cluster.
- If after replacing the meta variables with the concrete variables, an inconsistency happens $\Longrightarrow$

Choose the next best ECU by using the nogoods provided by the consistency checker

Conclusions

- (CSP) representations are successfully used in diverse areas from configuration and reconfiguration of large systems.
- We outline a novel modeling approach that allows deployment of embedded automotive software.
- The Model-based approach allows for automatic deployment of software functions in a resource-constrained software system.
- Computation of a valid solution satisfying the outlined criteria by relying on standard CSP solvers.
- We do not generate all the cluster combinations but rely on the the first n, solutions that the CSP solver comes up with.

Thank You!

Q & A